ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Algorithms for Radio Networks

Network Coding

University of Freiburg
Technical Faculty
Computer Networks and Telematics
Christian Schindelhauer

CoNe Freiburg

IIF

Data Flows in Networks

- **Motivation**
  - Optimize data flow from source to target
- **Definition:**
  - (Single-commodity) maximum flow problem
  - Given
    - a graph G=(V,E)
    - a capacity function $w: E \to R^{+}_{0}$,
    - source set S and target set T
  - Find a maximum flow from S to T

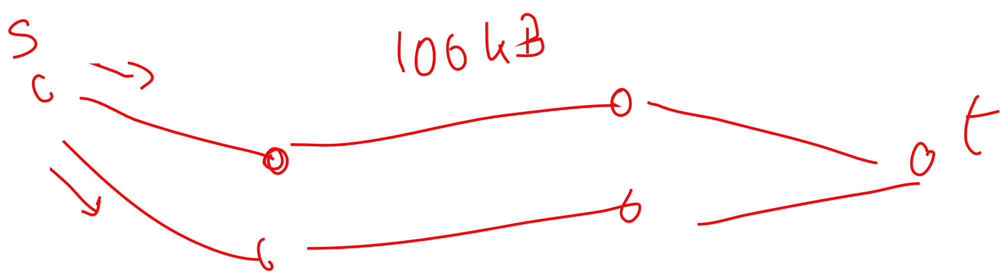
- **A flow is a function $f : E \to R_0^+$ such that**
  - for all $e \in E$: $f(e) \leq w(e)$
  - for all $e \notin E$: $f(e) = 0$
  - for all $u,v \in V$: $f(u,v) \geq 0$

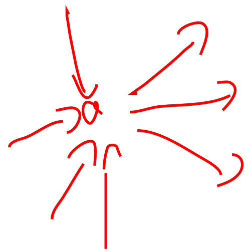
$$\forall u \in V \setminus (S \cup T)$$

$$\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v)$$

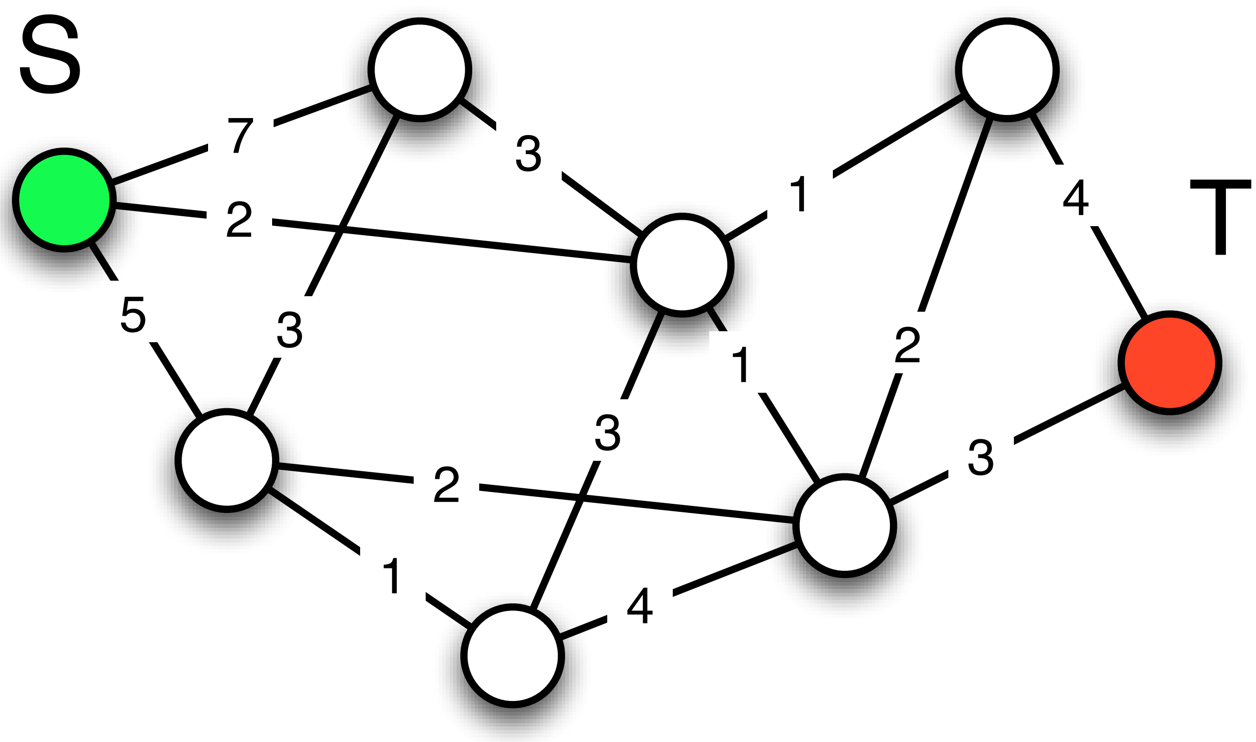
- **Maximize flow**

$$\sum_{u \in S} \sum_{v \in V} f(u,v)$$

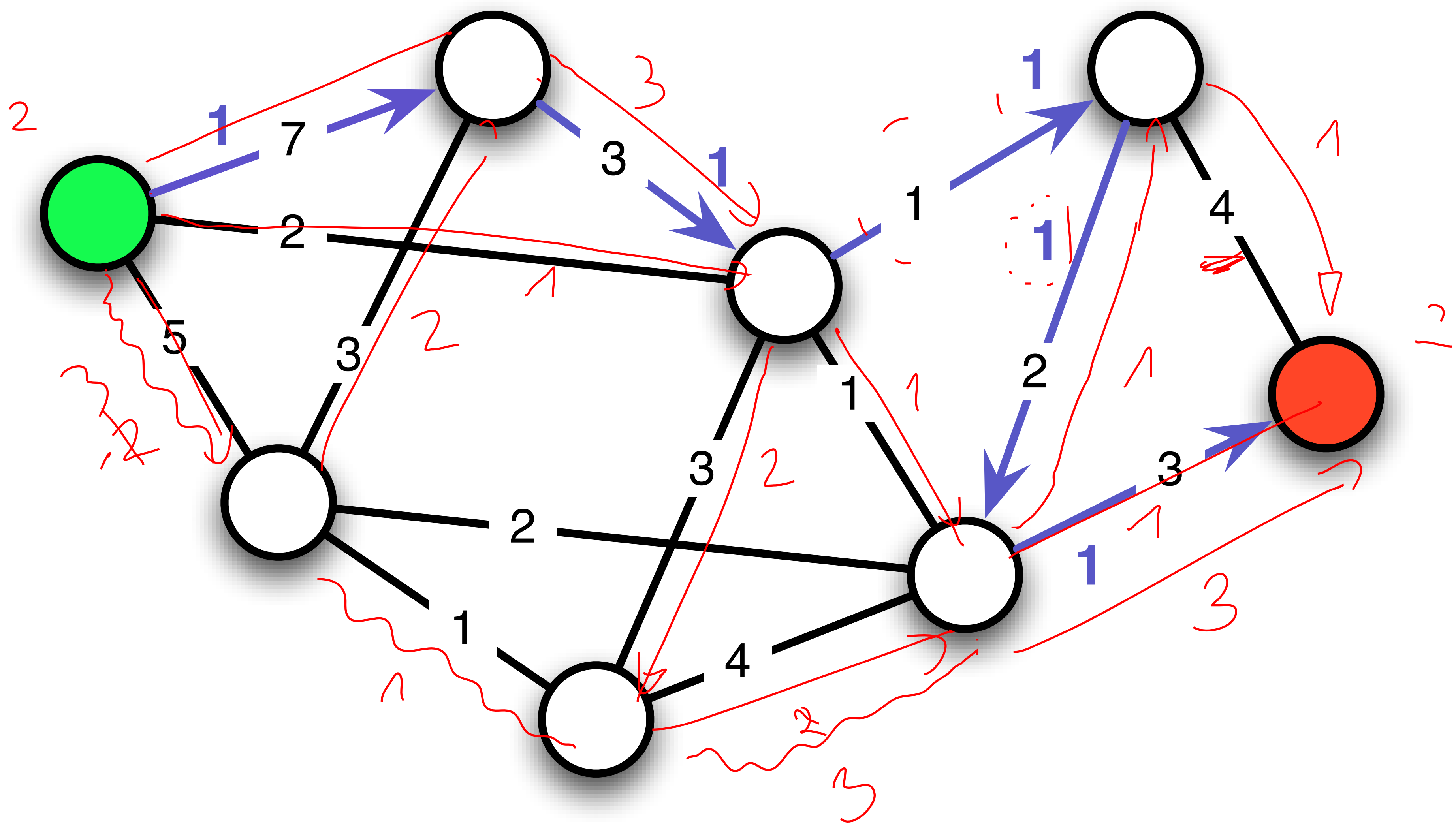




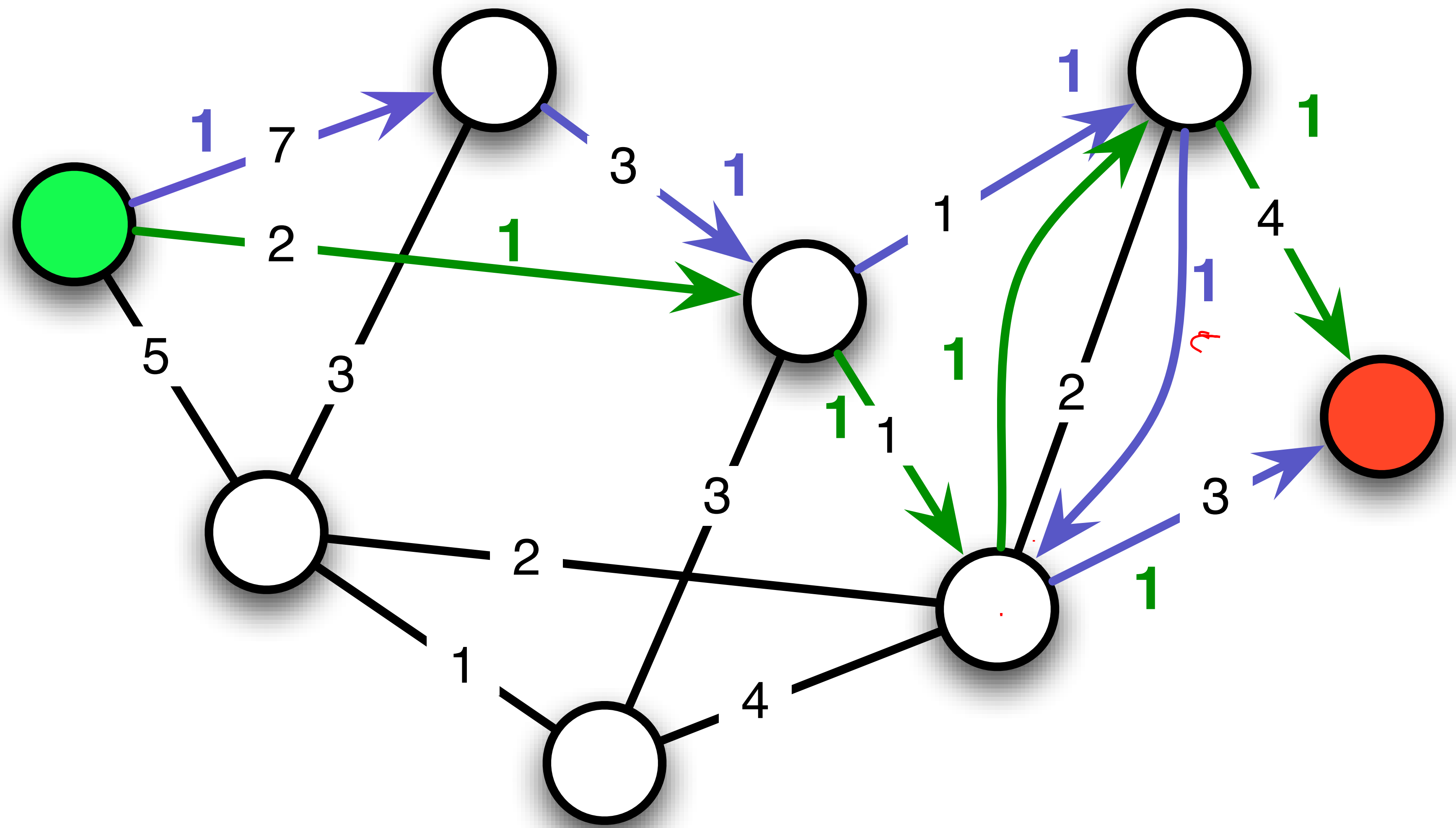
Data Flows in Networks



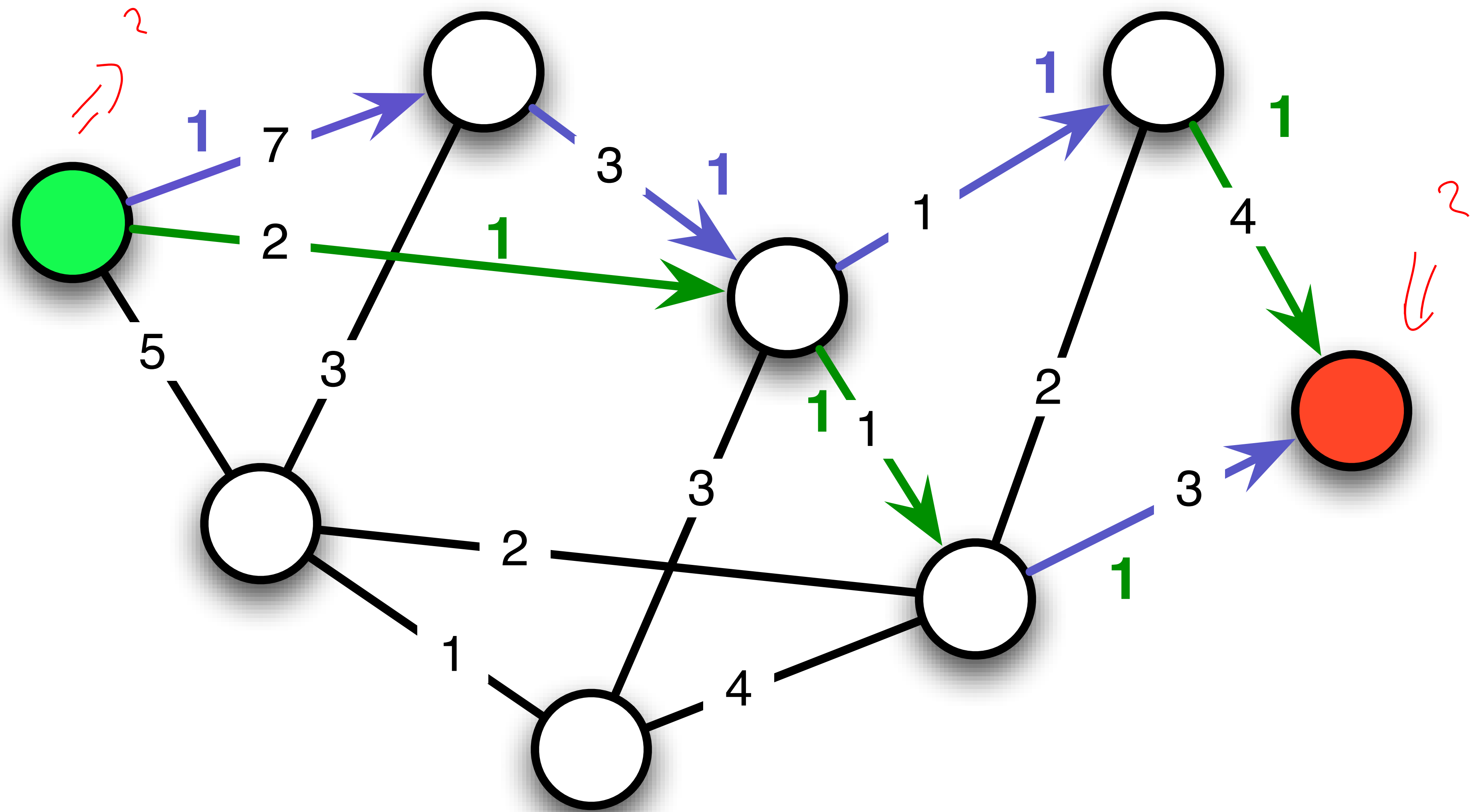
Data Flows in Networks



Data Flows in Networks

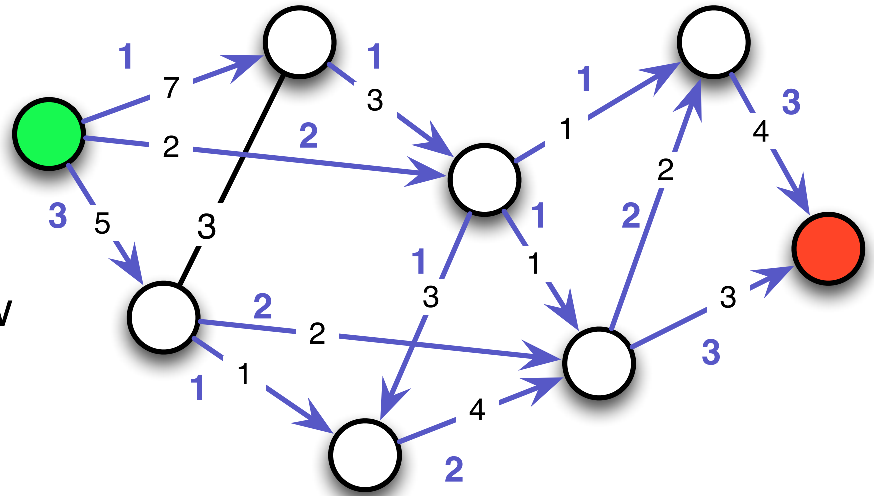


Data Flows in Networks



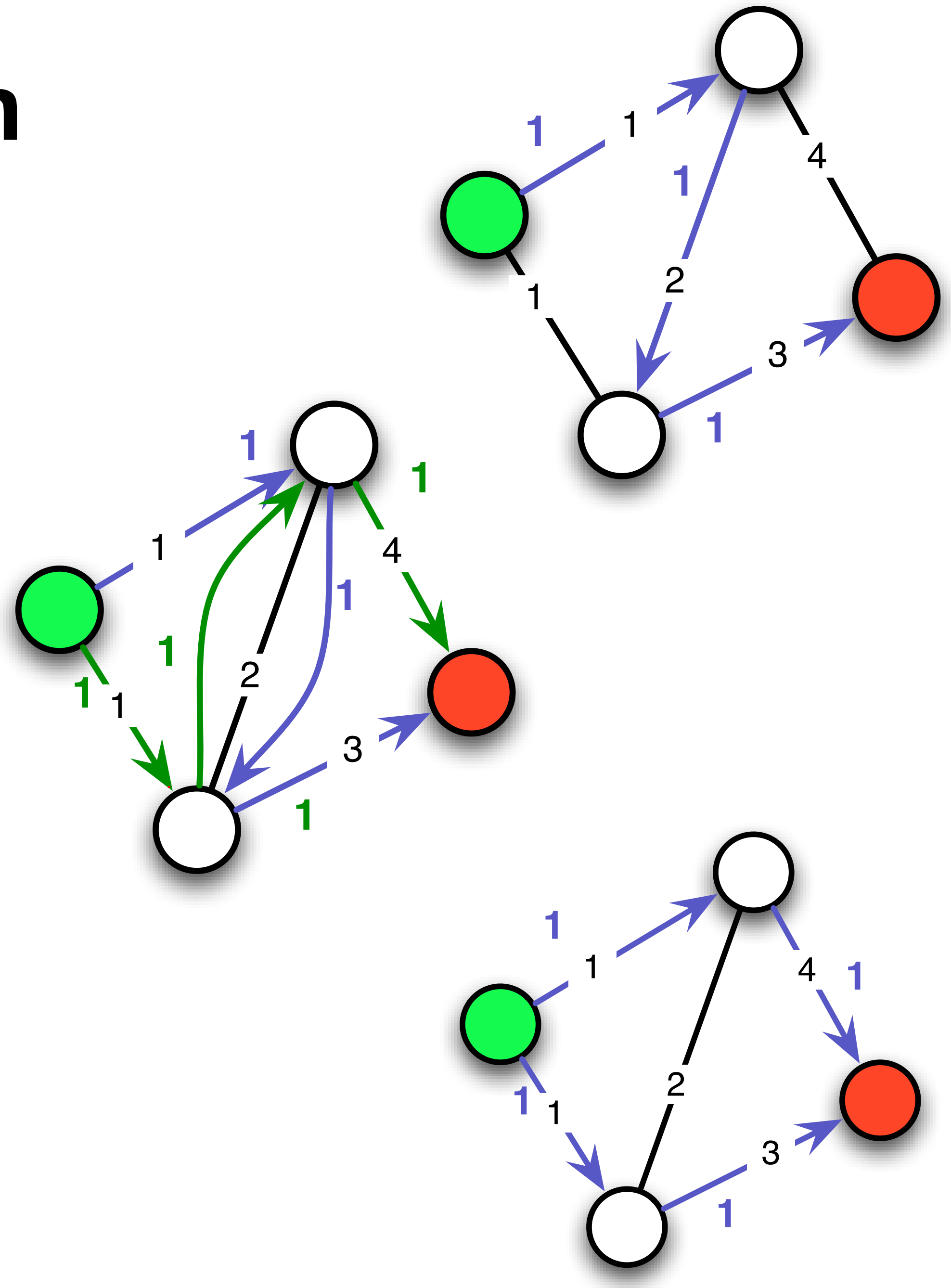
Computation of the Maximum Flow

- **Every natural pipe system solves the minimummaximum flow problem**
- **Algorithms**
  - Linear Programming
    - for real numbers
    - the flow is described by equations of a linear optimization problem
    - Simplex algorithm (or Ellipsoid method) can solve any linear equation system
  - Ford-Fulkerson
    - also for integers
    - as long as open paths exist, increase the flow on theses paths
      - open path: path which increases the flow
  - Edmonds-Karp
    - special case of Ford-Fulkerson
    - use BFS (breadth first search) to find open paths



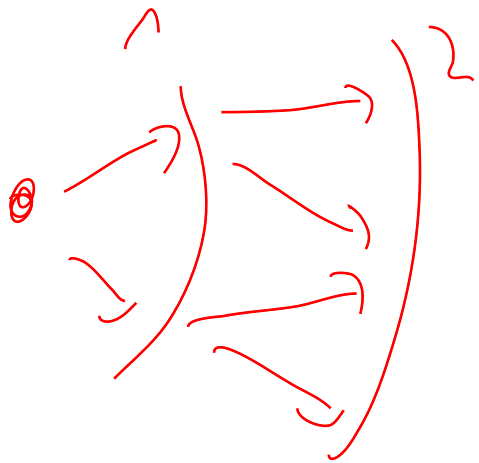
Ford-Fulkerson

- **Find a path from the source node to the target node**
  - where the capacity is not fully utilized
  - or which reduces the existing flow
- **Compute the maximum flow on this augmenting path**
  - by the minimum of the flow that can be added on all paths
- **Add the flow on the path to the existing flow**
- **Repeat this step until no flow can be added anymore**

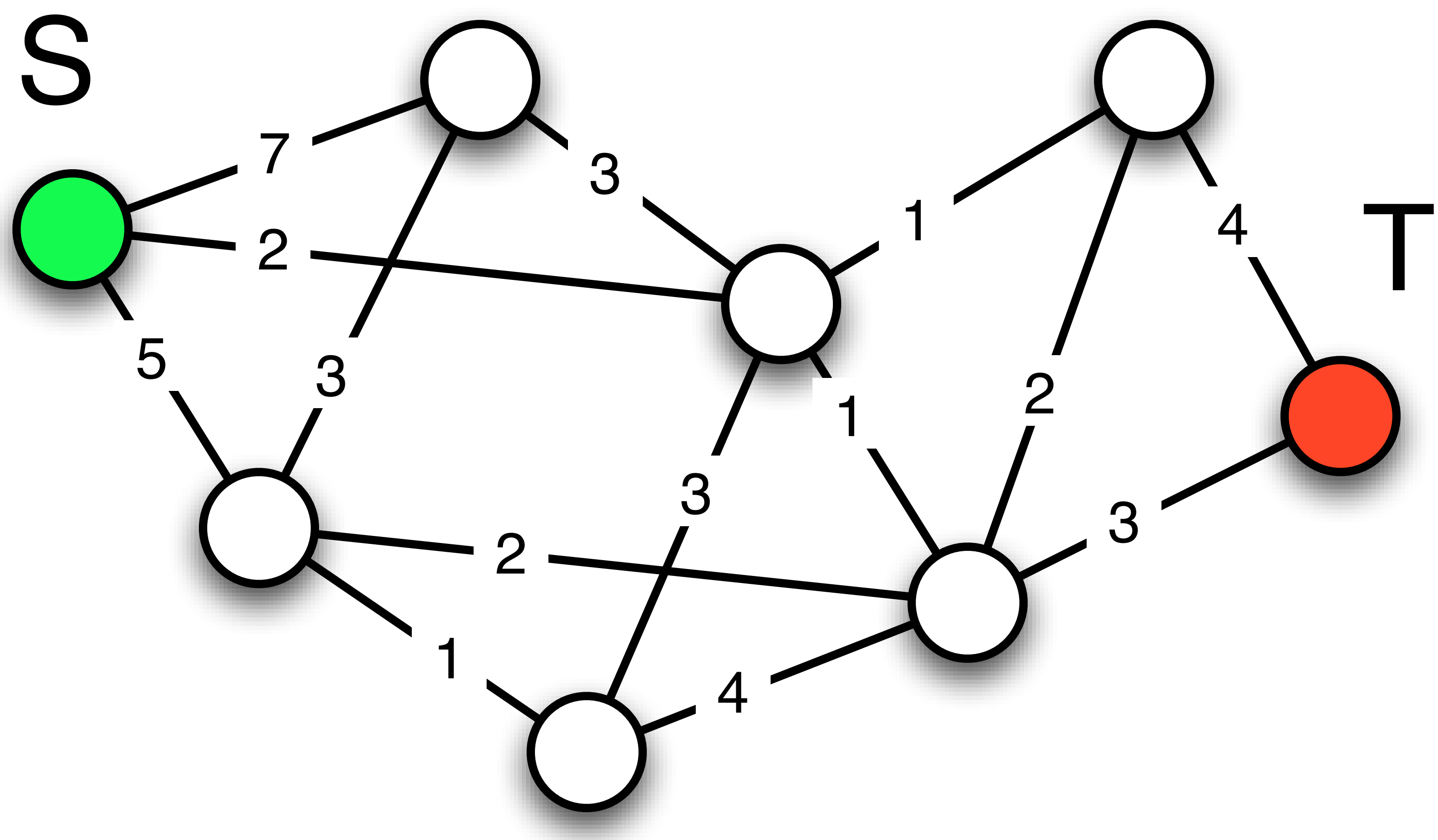


Edmunds-Karp

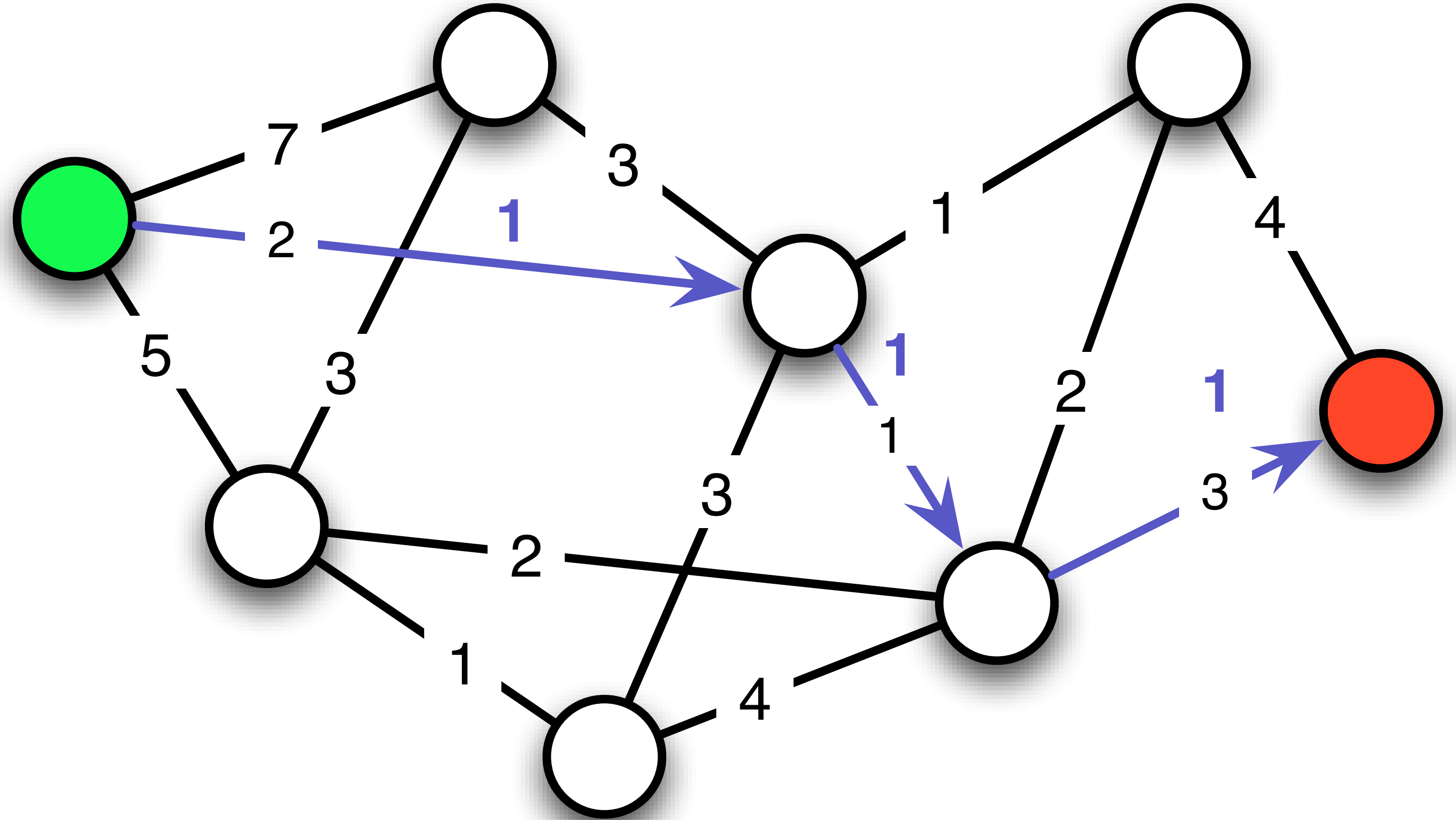
- **Search path for Ford-Fulkerson algorithm**
- **Choose the shortest augmenting path**
  - Computation by breadth-first-search
- **leads to run-time $O(|V|\ |E|^2)$**
  - whereas Ford-Fulkerson could have exponential run-time



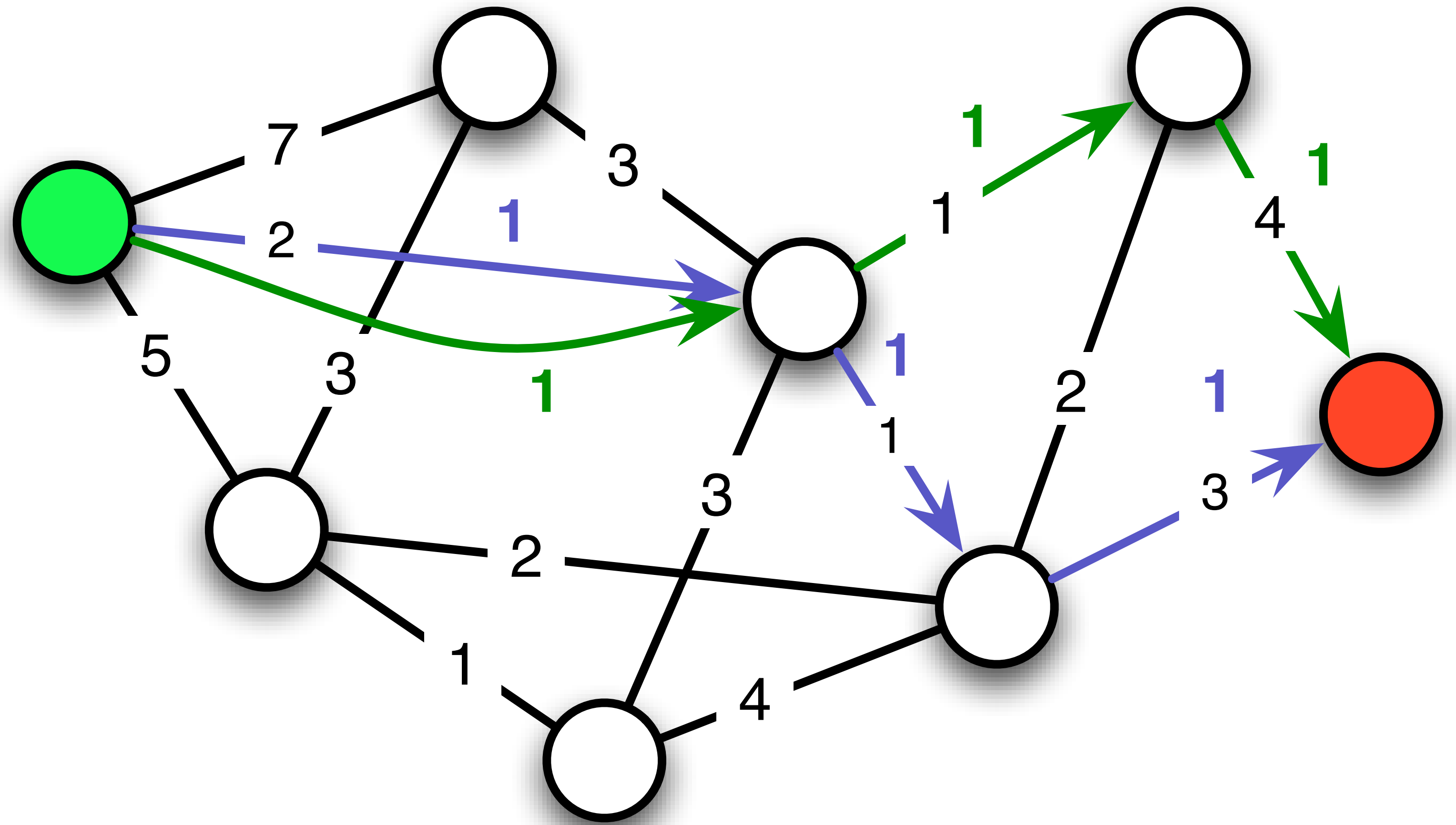
Example



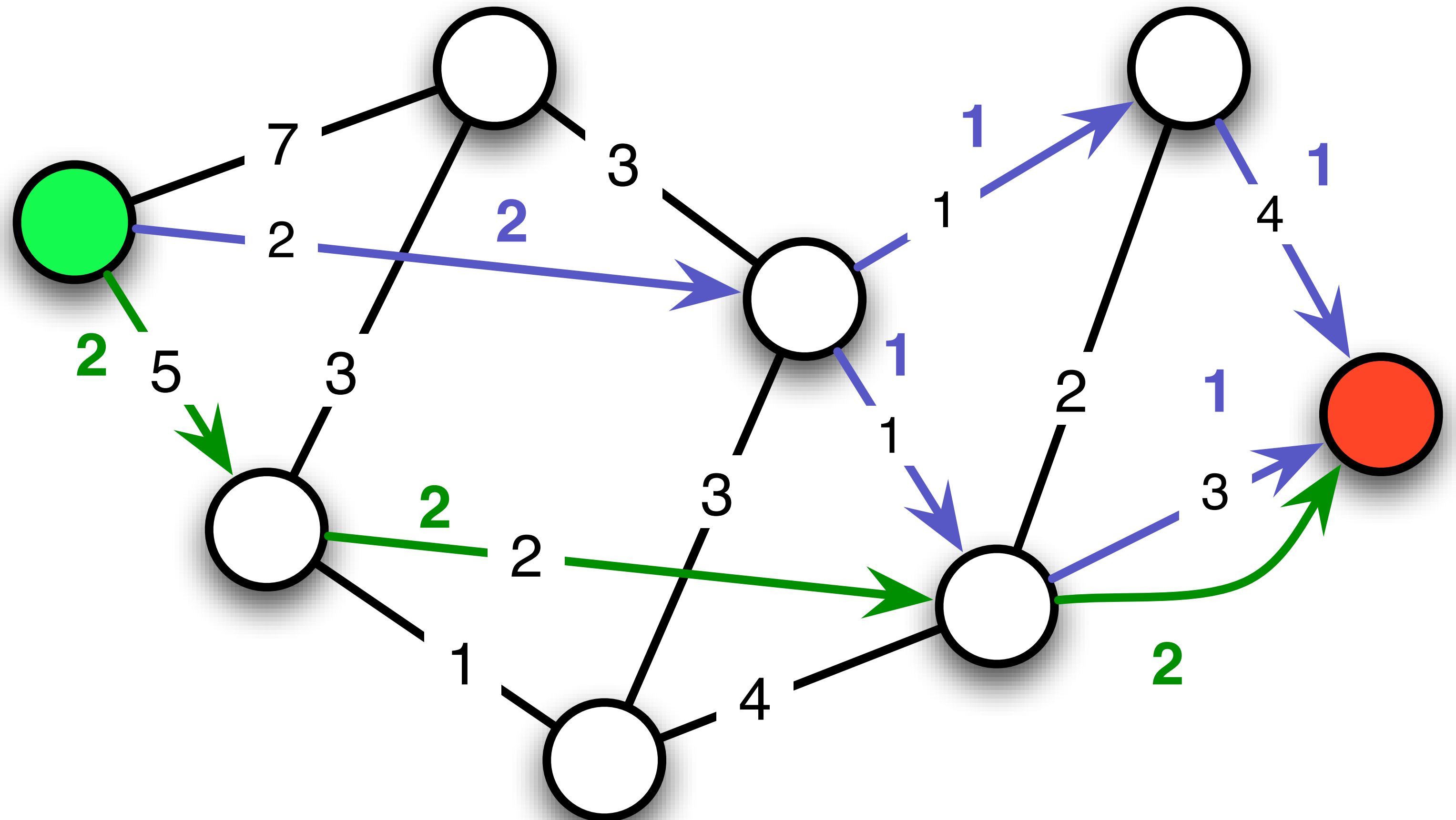
Example



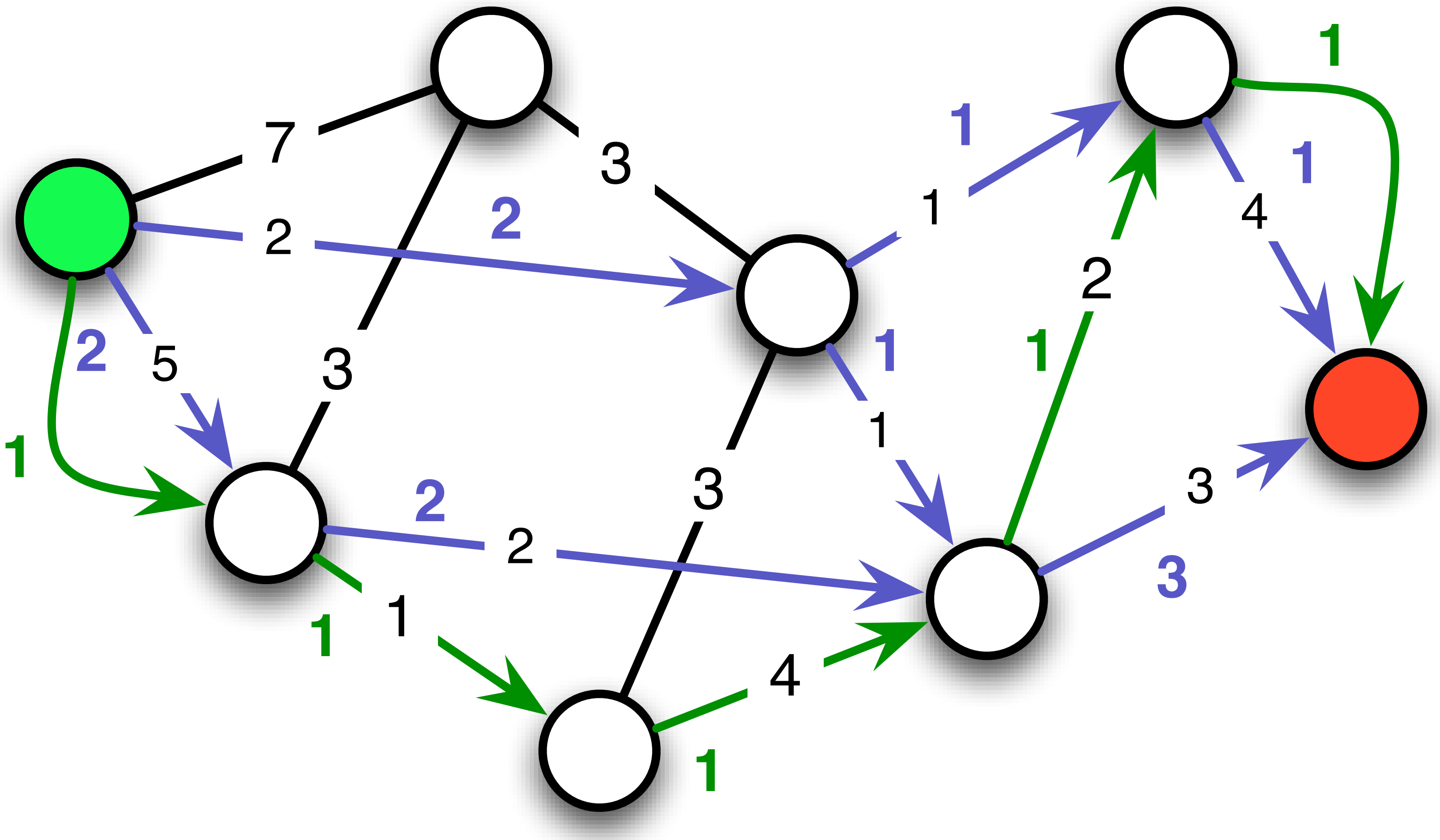
Example



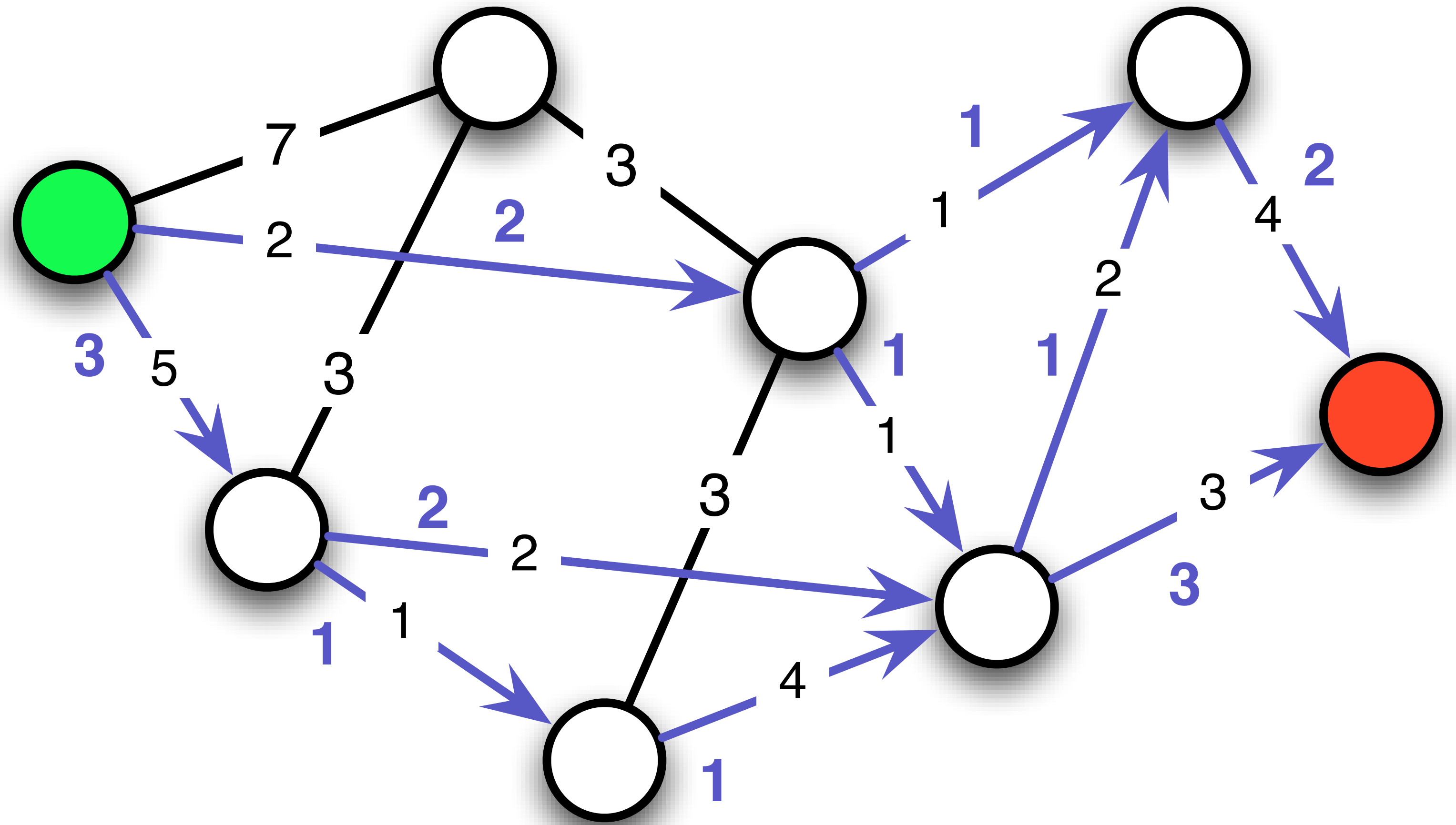
Example



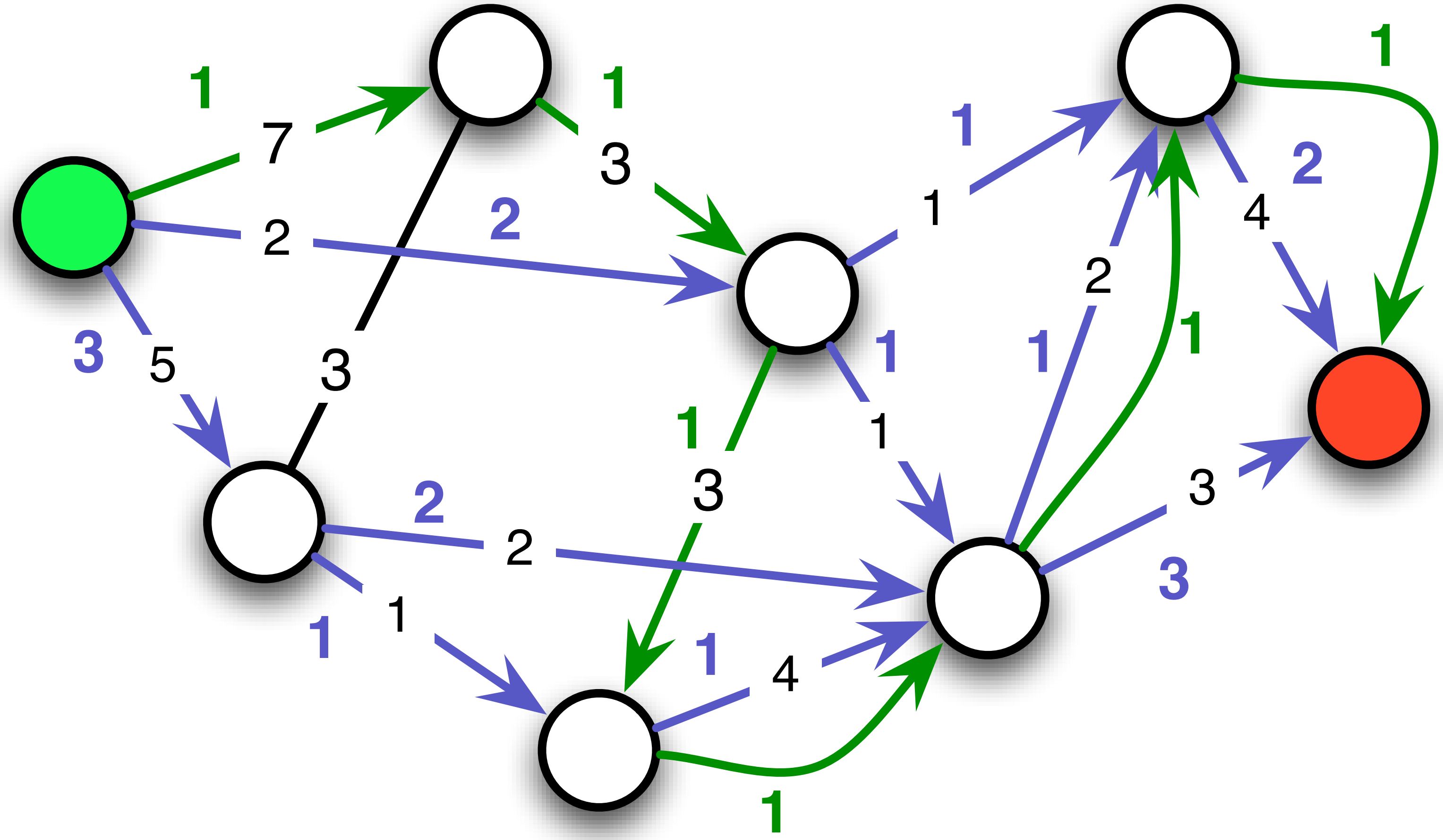
Example



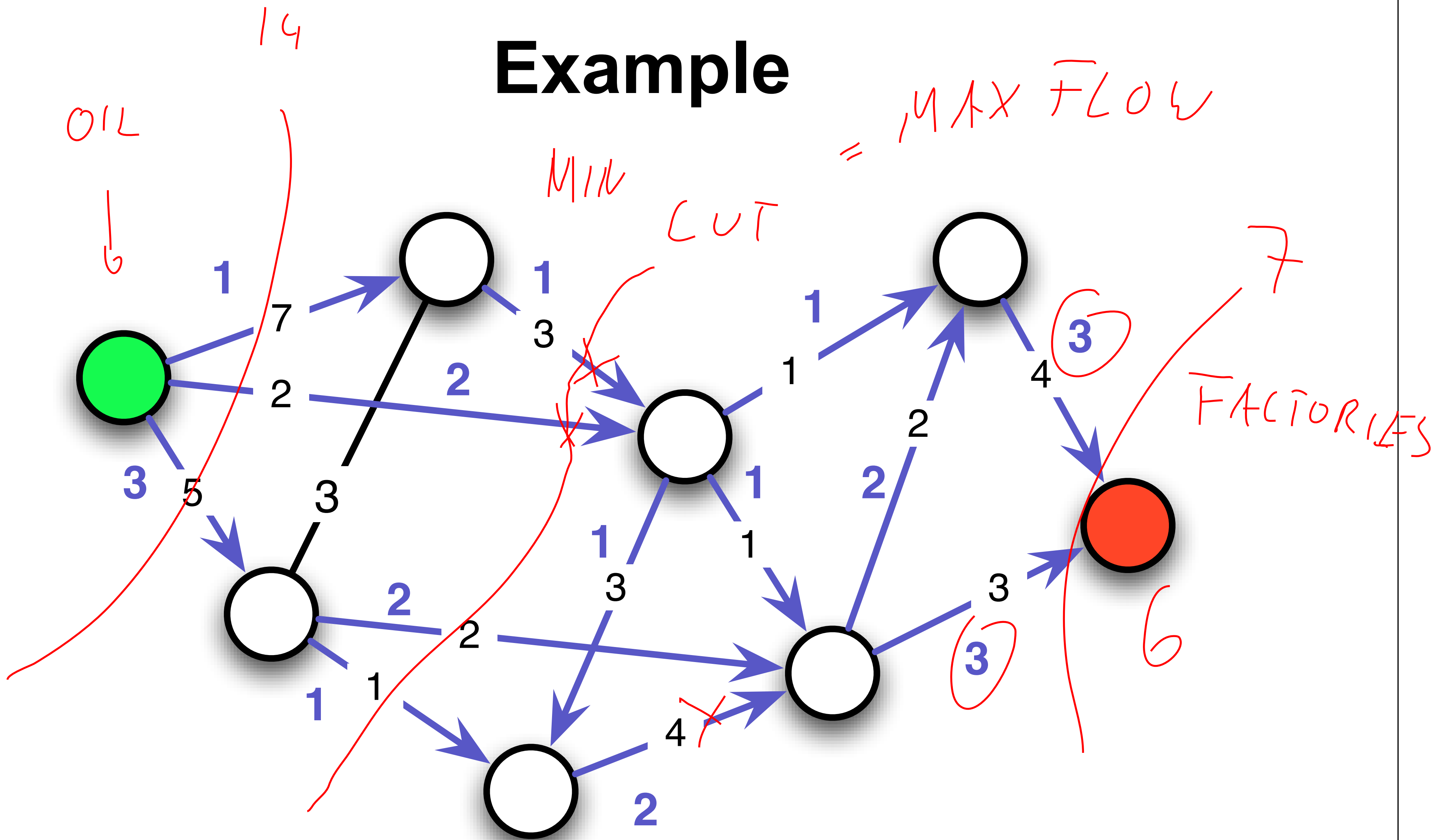
Example



Example



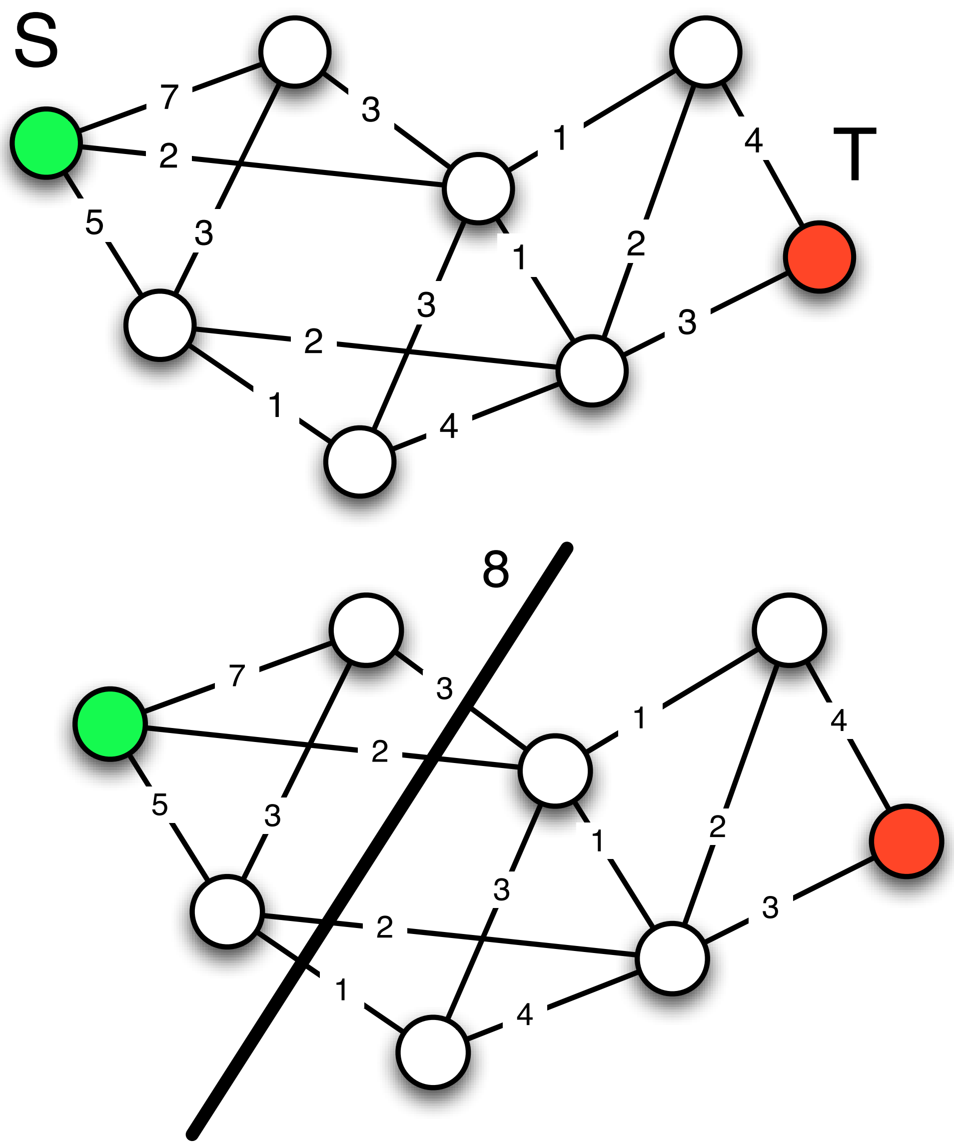
Example


14
OIL
6
MIN
CUT
= MAX FLOW
7
FACTORIES
6
1
7
2
2
3
5
3
1
3
2
2
1
1
1
2
1
1
3
1
3
1
2
2
2
4
3
4
3
3
2

Minimum Cut in Networks

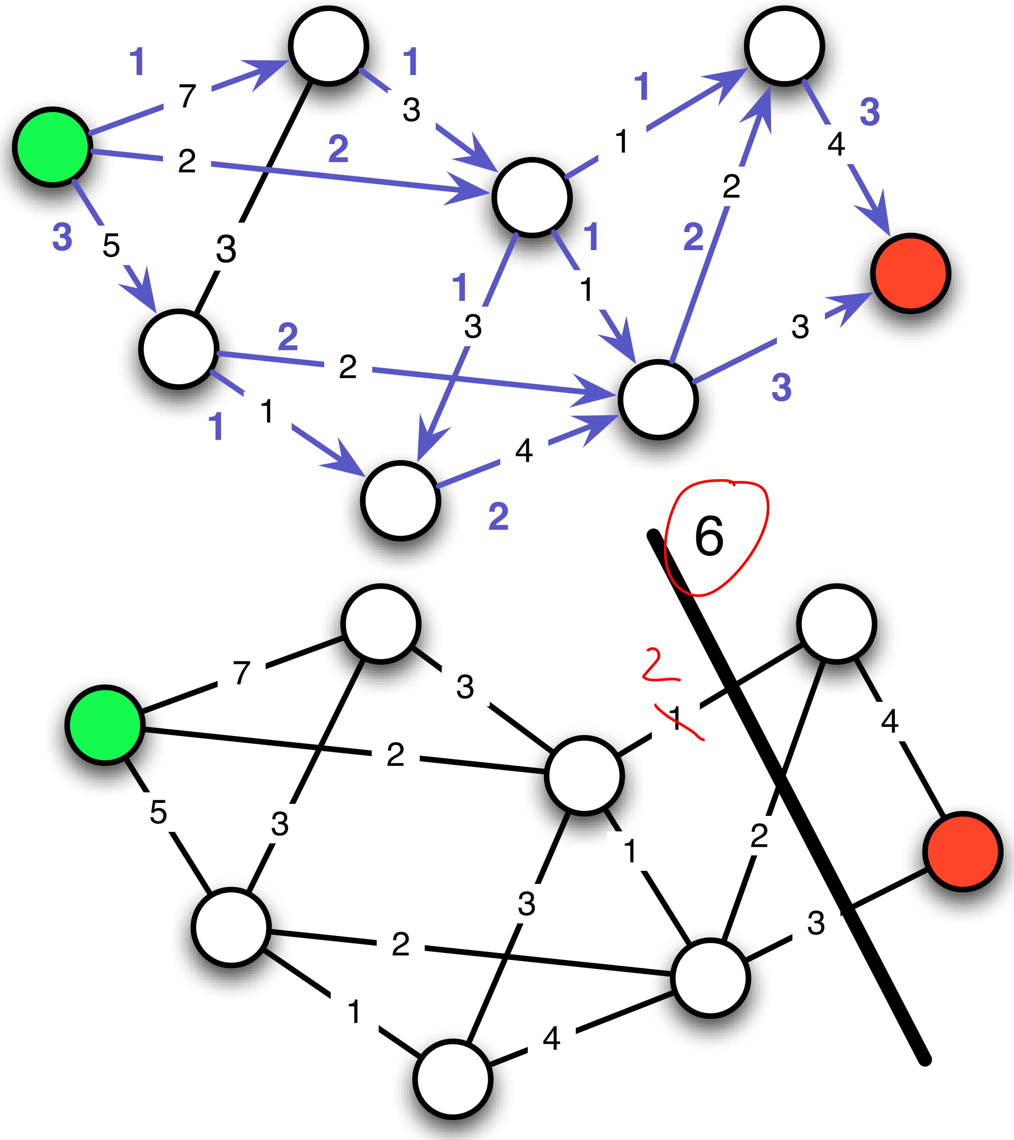
- **Motivation**
  - Find bottleneck in networks
- **Definition**
  - Min Cut problem
  - Given
    - graph G=(V,E)
    - capacity function w: E → R+0,
    - sources S and targets T
  - Find minimum cut between S and T
- **A cut C is a set of edges**
  - such that every path from a node of S to a node of T, contains an edge of C
- **The size of a cut is**

$$\sum_{e \in C} w(e)$$



Min-Cut-Max-Flow Theorem

- **Theorem**
  - The minimum cut equals the maximum flow
- **Algorithms for minimum cut**
  - can be obtained from the maximum flow algorithms



Multi-Commodity Flow Problem

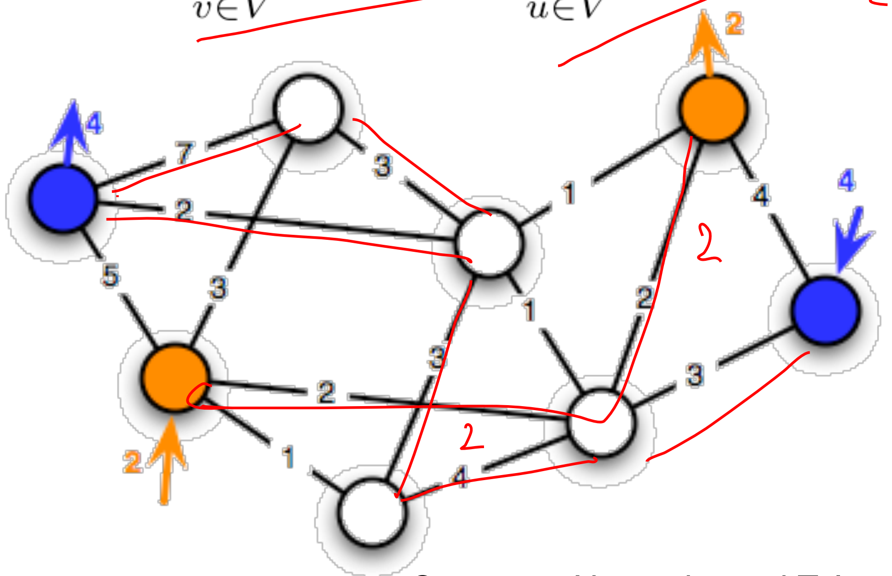
- **Motivation**
  - theoretical model for point to point communication
- **Definition**
  - Multi-commodity flow problem
  - given
    - a graph G=(V,E)
    - a capacity function w: E → R+0,
    - commodities K_1, .., K_k:
      - $K_i=(s_i,t_i,d_i)$ with
      - s_i: source node
      - t_i: target node
      - d_i: demand

- **Find flows $f_1,f_2,\ldots,f_k$ for all commodities such that**
  - capacities
  - flow property
  - demand

$$\sum_{i=1}^{k} f_i(u,v) \leq w(u,v)$$

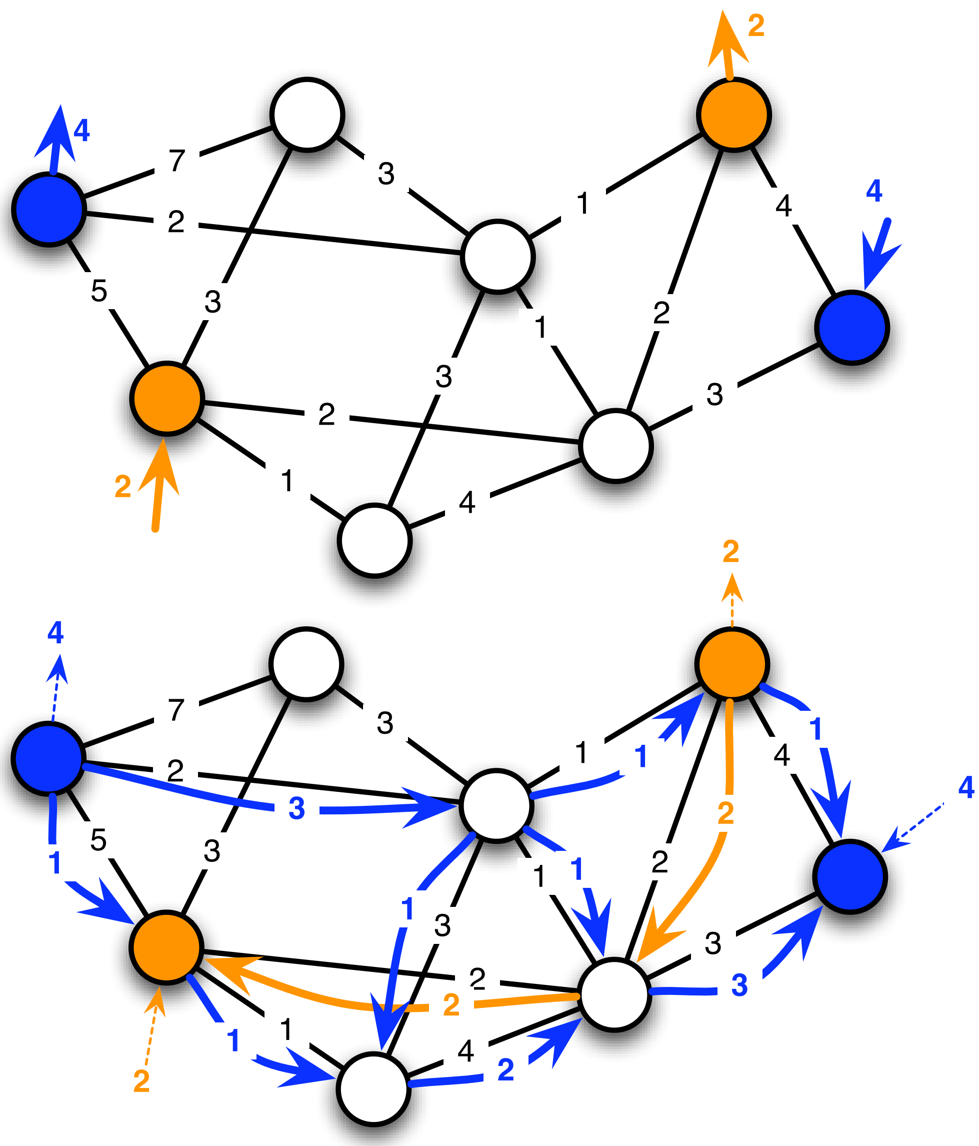
$$\forall v \notin \{s_i,t_i\} \ : \ \sum_{u\in V} f_i(u,v) = \sum_{u\in V} f_i(v,u)$$

$$\sum_{v\in V} f_i(s_i,v) = \sum_{u\in V} f_i(u,t_i) = d_i$$



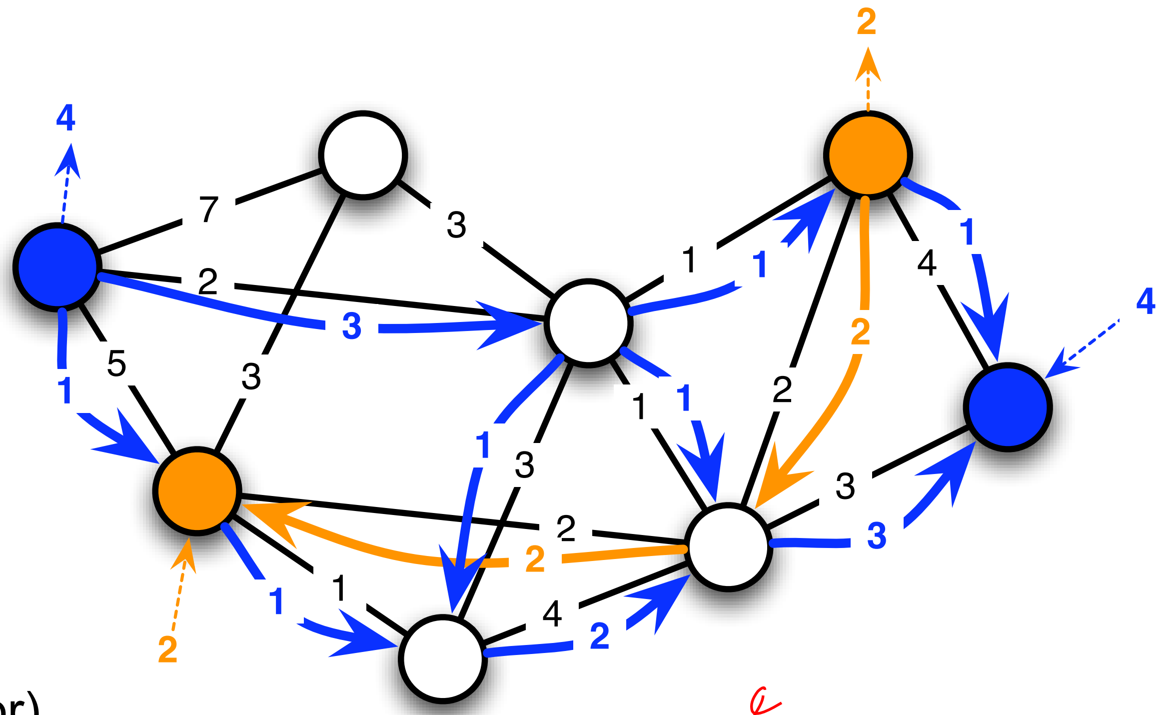
Solving the Multi-Commodity Flow Problem

- **Multi-Commodity Flow Problem**
- **Optimize**
  - sum of all flows or
  - maximize the worst ratio between commodity and the demand
- **Problem can be solved in polynomial time**
  - for real numbers
  - using linear programming



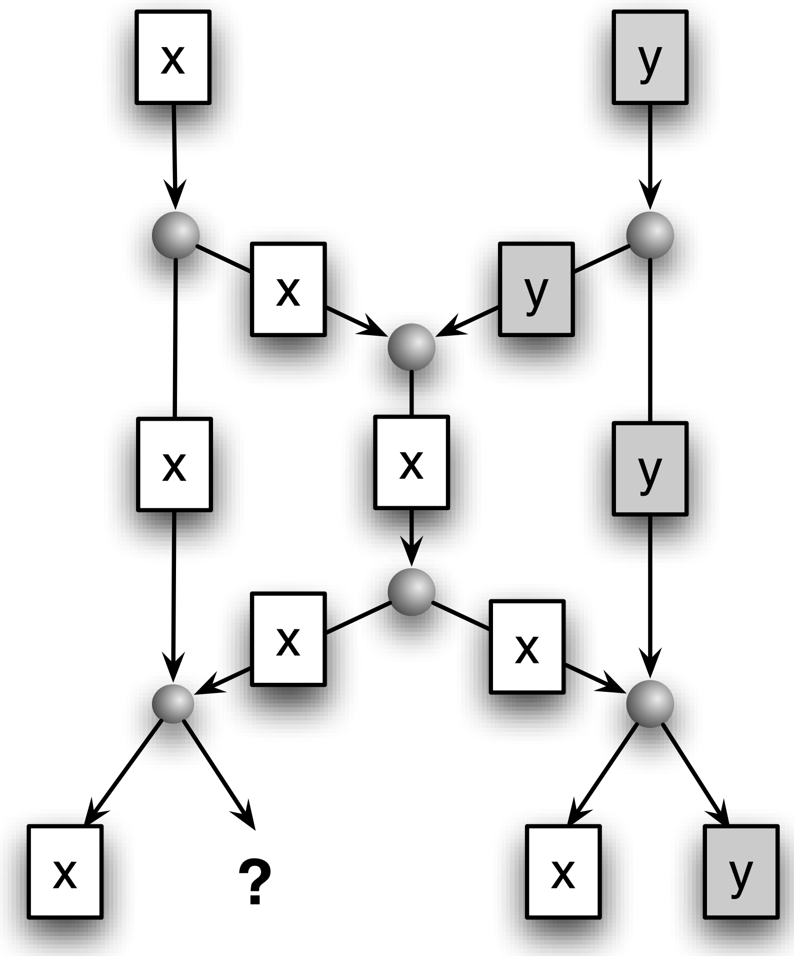
Complexity of the Multi Commodity Flow Problem

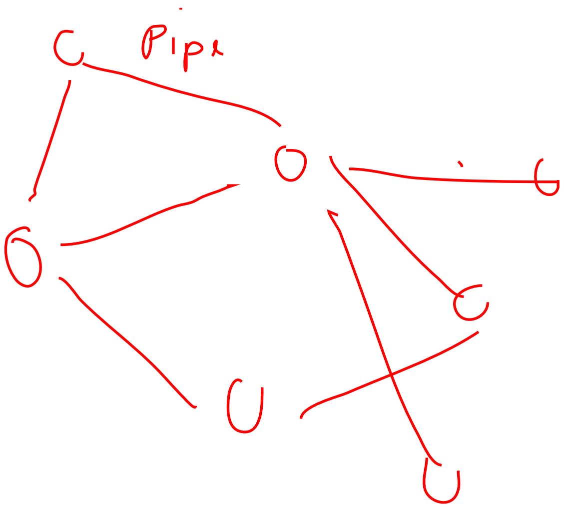
- **Problem is NP-complete**
  - for integers
    - e.g. packets
  - even for two commodities
    - Shai, Itai, Even, 1976
- **Polynomial solution**
  - with respect to the number of paths between sources and targets
- **Approximation**
  - good central and distributed approximation algorithms exist (polylogarithmic approximation factor)
- **Weaker forms of the Min-Cut-Max-Flow-Theorems exist**



Network Coding

- **R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung**
  - *Network Information Flow*, (IEEE Transactions on Information Theory, IT-46, pp. 1204-1216, 2000)
- **Example**
  - Bits x and y are to be transfered
  - Each edge carries only a bit
  - If bits are transfered as is
    - then both x and y cannot be received either on the left or right side





→ TCP/IP

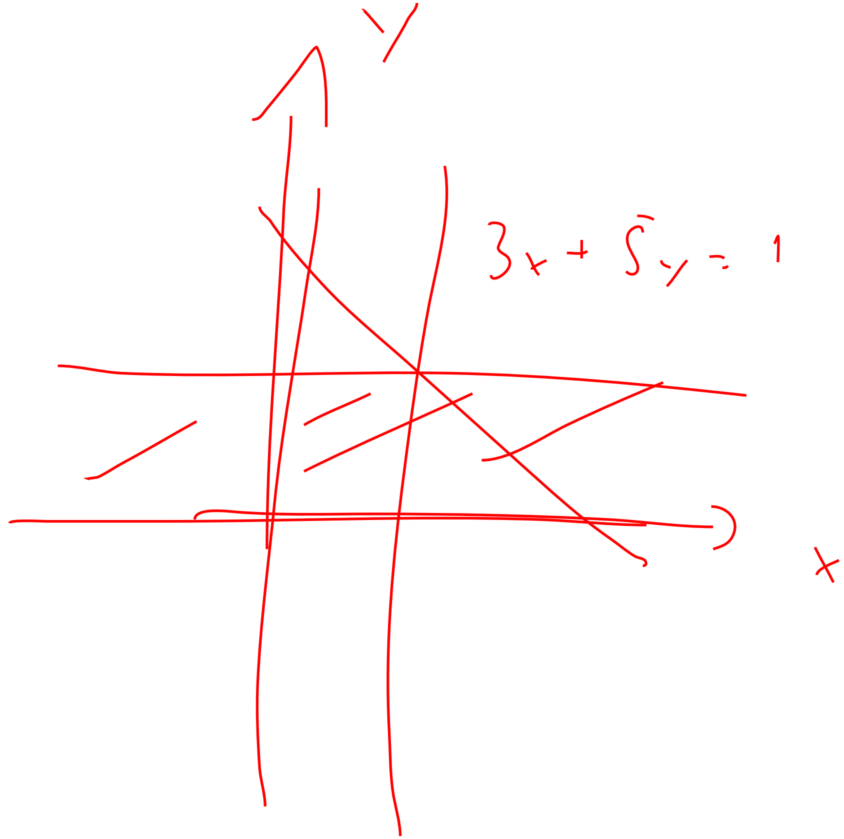
→ Flow problem

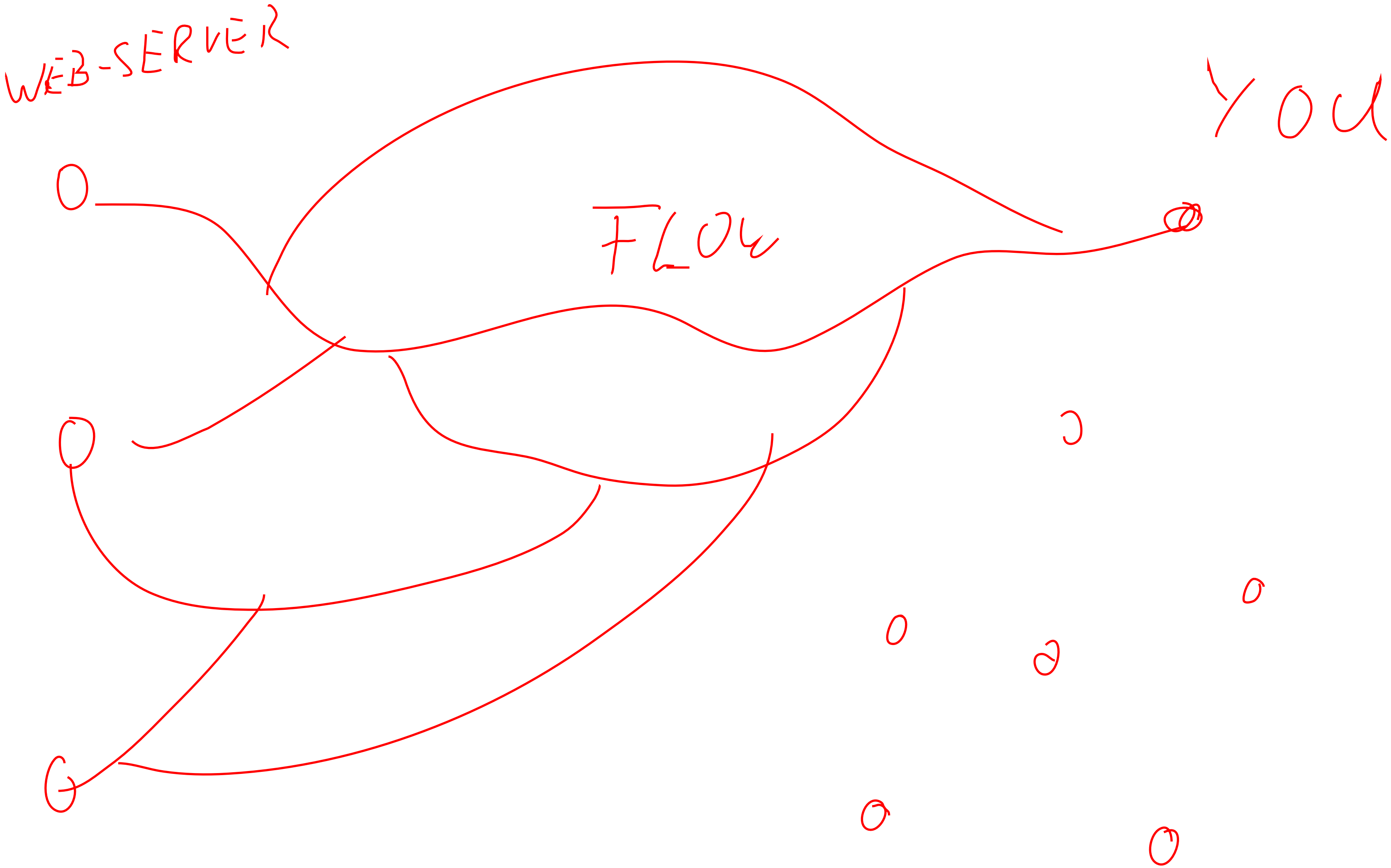
$$\boxed{3x + 5y = 1}$$

$x \leq 10$ $\qquad$ $x \geq 0$

$y \leq 15$ $\qquad$ $y \geq 0$

$$\arg\max_{x,y} \boxed{2x - y}$$




WEB-SERVER
FLOW
YOU